

Amendments to the Claims:

What is claimed is:

1. (Currently Amended) A method for loading a portable executable (PE) image, the method comprising:
 - determining whether a PE image for a platform firmware runtime service includes a discardable section;
 - loading part of the PE image into runtime memory to be used exclusively by the platform firmware; and
 - in response to determining that the PE image includes a discardable section, omitting at least part of the discardable section when loading the PE image into the runtime memory.
2. (Original) A method according to claim 1, further comprising:
 - loading the discardable section into boot-time memory to be used by the platform firmware.
3. (Original) A method according to claim 1, further comprising:
 - using an alignment granularity of less than one kilobyte when loading the PE images into the runtime memory.
4. (Original) A method according to claim 1, further comprising:
 - using an alignment granularity of less than one hundred bytes when loading the PE images into the runtime memory.
5. (Original) A method according to claim 1, further comprising:
 - pre-allocating an area of runtime memory for PE images; and
 - loading sections from multiple PE images into the pre-allocated area of runtime memory.
6. (Original) A method according to claim 1, further comprising:
 - pre-allocating an area of runtime memory for PE images; and

loading sections from multiple PE images into the pre-allocated area of runtime memory;
and

using an alignment granularity of less than one kilobyte when loading the PE images into the pre-allocated area of runtime memory.

7. (Original) A method according to claim 1, further comprising:
recording a runtime memory size in association with a first boot process; and
pre-allocating an area of runtime memory for PE images in association with a subsequent boot process, based at least in part on the recorded runtime memory size.

8. (Original) A method according to claim 1, further comprising:
recording a runtime memory size in association with a first boot process;
pre-allocating an area of runtime memory for PE images in association with a subsequent boot process, based at least in part on the recorded runtime memory size; and
loading sections from multiple PE images into the pre-allocated area of runtime memory.

9. (Original) A method according to claim 1, further comprising:
recording a first runtime memory size in association with a first boot process;
pre-allocating an area of runtime memory for PE images in association with a subsequent boot process, based at least in part on the recorded runtime memory size;
loading sections from multiple PE images into the pre-allocated area of runtime memory;
determining how much of the pre-allocated area of runtime memory was used; and
recording a second runtime memory size in association with the second boot process,
based at least in part on the determination of how much of the pre-allocated area of runtime memory was used.

10. (Original) A method according to claim 1, wherein the PE image comprises header information, the method further comprising:
omitting at least part of the header information when loading the PE image into the runtime memory.

11. (Currently Amended) A method for creating a portable executable (PE) image, the method comprising:

receiving an object file at a linker, the object file containing multiple discardable sections with instructions for performing boot-time operations and a section with instructions for performing runtime operations to be provided by platform firmware; and

generating an executable PE image, based at least in part on the object file;

wherein the operation of generating the executable PE image comprises grouping the multiple discardable sections together in the PE image.

12. (Original) A method according to claim 11, wherein the operation of grouping the multiple sections with instructions for performing boot-time operations together in the PE image comprises:

grouping the multiple discardable sections together below the section with instructions for performing runtime operations.

13. (Currently Amended) A method for booting a processing system, the method comprising: retrieving a portable executable (PE) image for a runtime service to be provided by platform firmware for the processing system;

determining whether the PE image includes a discardable section;

in response to determining that the PE image includes a discardable section, loading the discardable section into boot-time memory to be used by the platform firmware; and

loading part of the PE image into runtime memory to be used exclusively by the platform firmware; but

omitting at least part of the discardable section when loading the PE image into the runtime memory.

14. (Original) A method according to claim 13, further comprising:

pre-allocating an area of runtime memory for PE images;

loading sections from multiple PE images into the pre-allocated area of runtime memory;
and

using an alignment granularity of less than four kilobytes when loading the PE images into the pre-allocated area of runtime memory.

15. (Original) A method according to claim 13, wherein the PE image comprises header information, the method further comprising:

omitting at least part of the header information when loading part of the PE image into the runtime memory.

16. (Currently amended) An apparatus containing control logic for providing a runtime service for a processing system, the apparatus comprising:

a machine-accessible storage medium; and

a portable executable (PE) image in the machine-accessible storage medium, the PE image for providing a runtime service to be provided by platform firmware for the processing system, wherein the PE image comprises:

a section with instructions for performing runtime operations; and

multiple discardable sections with instructions for performing boot-time operations,

wherein the multiple discardable sections are grouped together in the PE image.

17. (Original) An apparatus according to claim 16, wherein the multiple discardable sections are grouped together below the section with instructions for performing runtime operations.

18. (Currently amended) A processing system with control logic for managing PE images, the processing system comprising:

a processor;

a machine-accessible storage medium responsive to the processor;

instructions in the machine-accessible storage medium which, when executed by the processor, implement an image loader; and

a portable executable (PE) image in the machine-accessible storage medium, the PE image for providing a runtime service in platform firmware for the processing system;

wherein the PE image comprises:

a section with instructions for performing runtime operations; and

multiple discardable sections with instructions for performing boot-time operations; and

wherein the multiple discardable sections are grouped together in the PE image.

19. (Currently Amended) A processing system according to claim 18, wherein the image loader comprises control logic:

to determine whether the PE image includes a discardable section;

to load part of the PE image into runtime memory to be used exclusively by the platform firmware; and

in response to determining that the PE image includes a discardable section, to omit at least part of the discardable section when loading the PE image into the runtime memory.

20. (Original) A processing system according to claim 19, wherein the image loader comprises control logic to load the discardable section into boot-time memory to be used by the platform firmware.

21. (Original) A processing system according to claim 19, wherein the image loader comprises control logic:

to pre-allocate an area of runtime memory for PE images;

to load sections from multiple PE images into the pre-allocated area of runtime memory;

and

to use an alignment granularity of less than four kilobytes when loading the PE images into the pre-allocated area of runtime memory.

Amendments to the Drawings:

Corrected drawings are supplied herewith, including one replacement sheet which include Fig. 2, and which replace the original sheet.

In Fig. 2 the branch from 240 to 244 was changed from “Yes” to “No.”

Attachment: One replacement sheet.

4